

Fig. 1

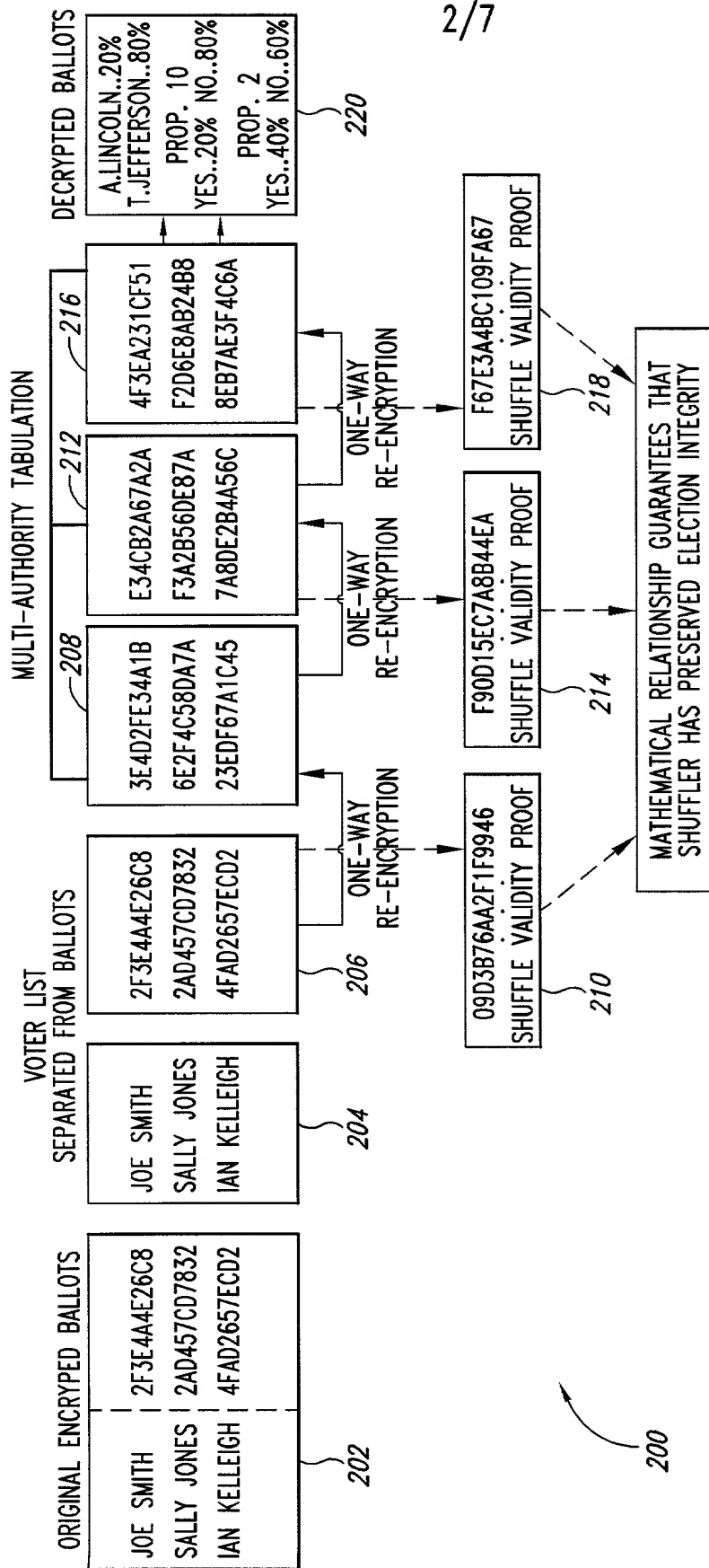


Fig. 2

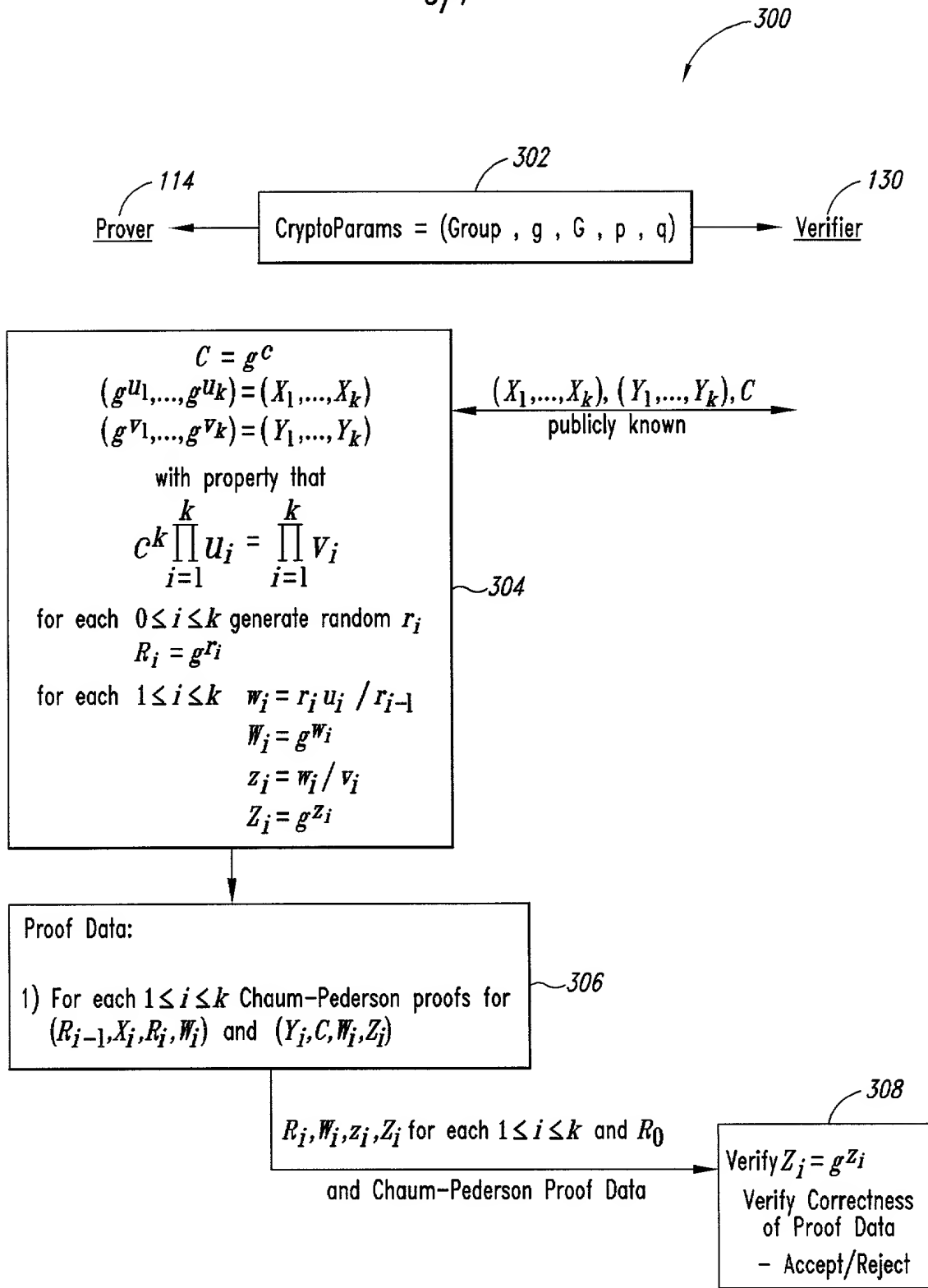


Fig. 3

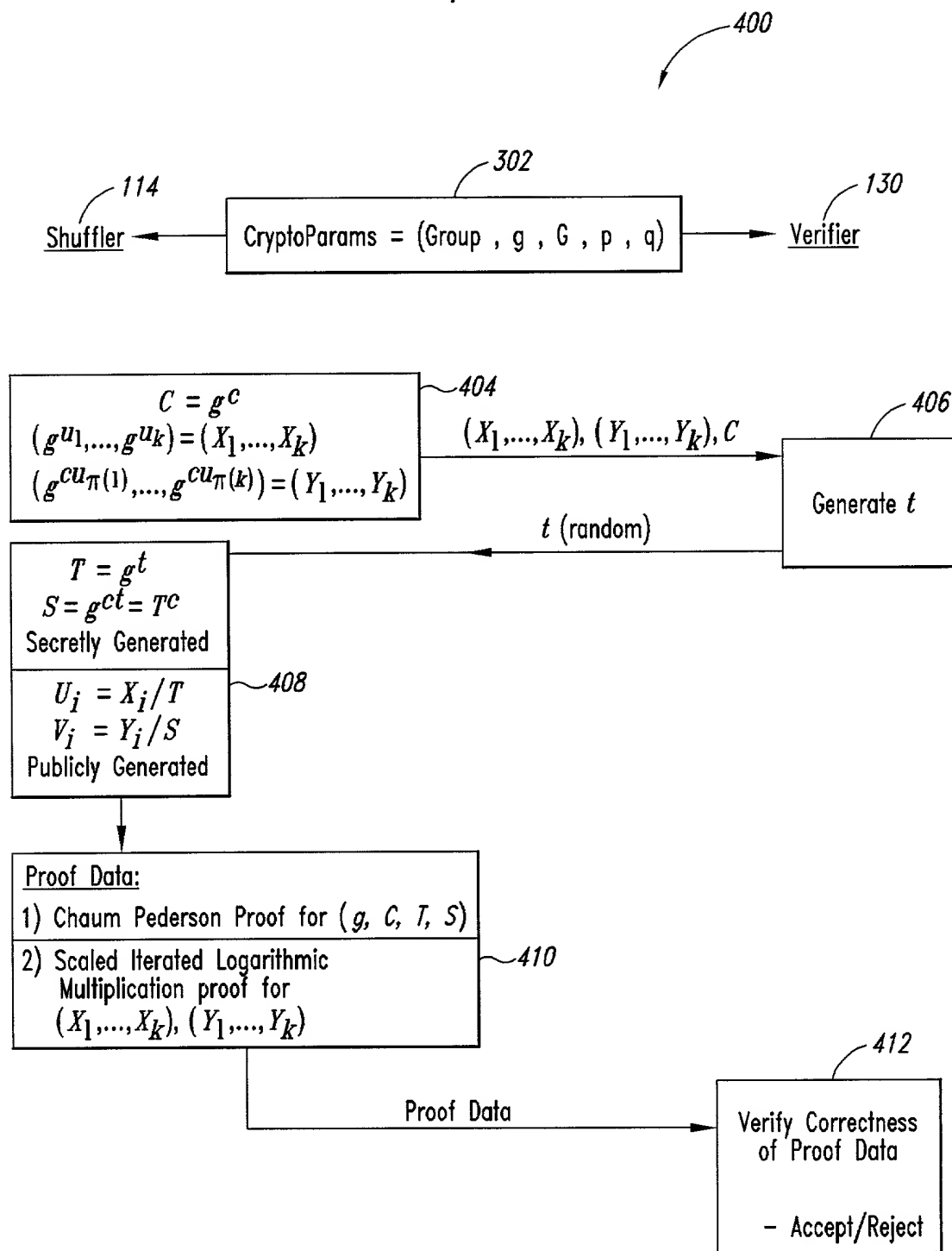
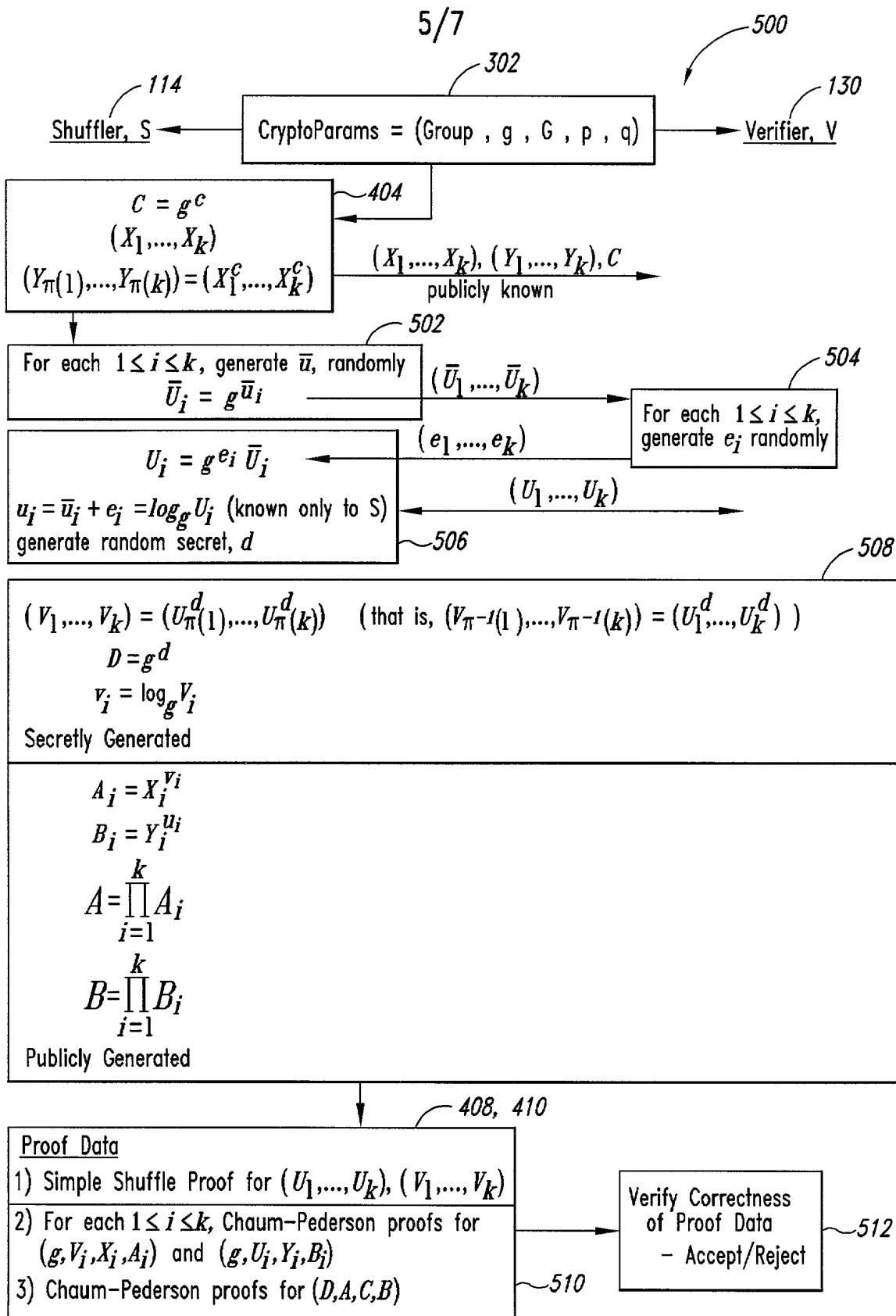


Fig. 4



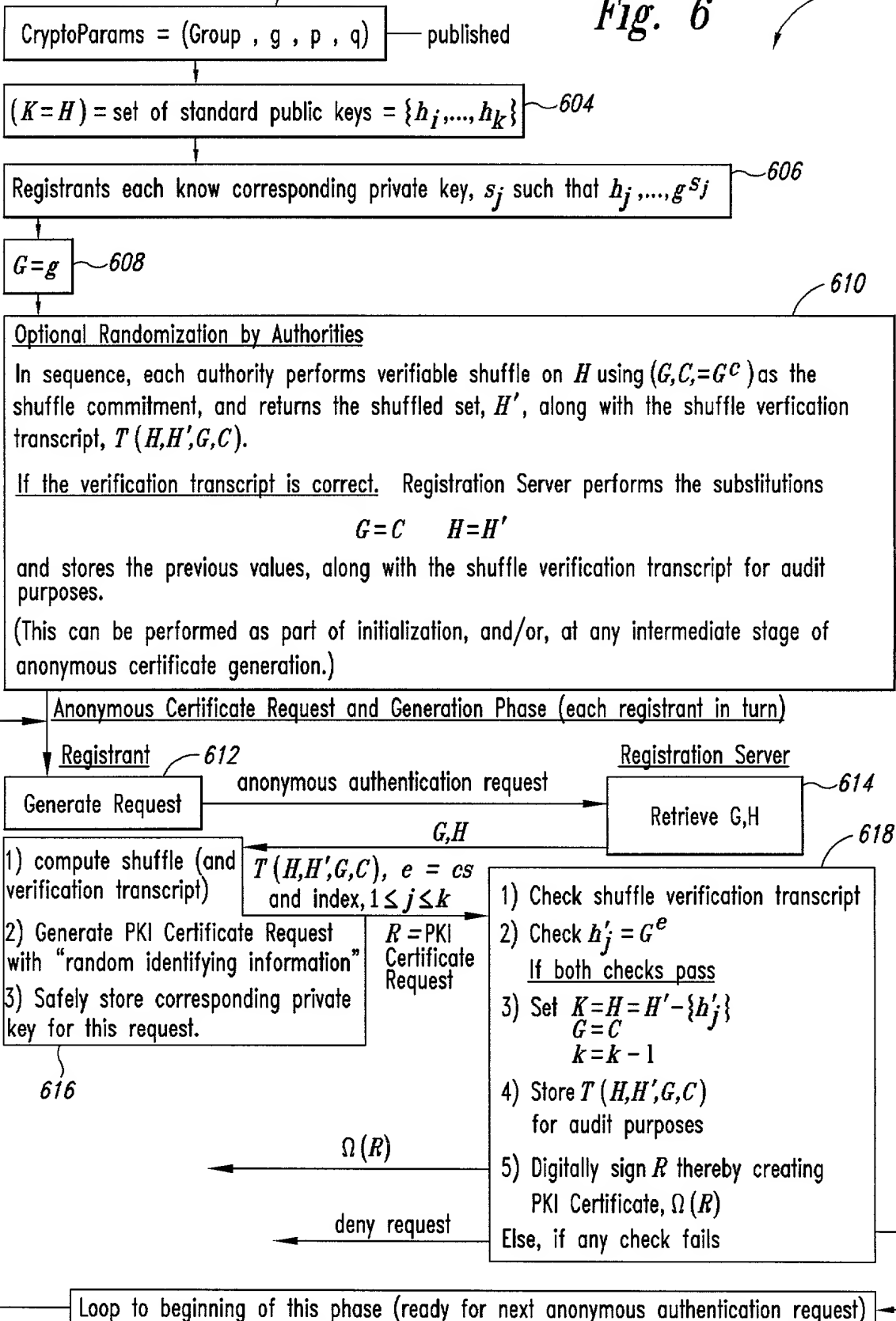
*Fig. 5*

# Initialization

6/7

Fig. 6

600



# Initialization

7/7

Fig. 7

700

CryptoParams = (Group,  $g$ ,  $p$ ,  $q$ ) — published

$K$  = set of standard public key pairs =  $\{(g_1, h_1), \dots, (g_k, h_k)\}$  704  
 $H \subset K$      $J = K - H$

Registrants each know corresponding private key,  $s_j$  such that  $h_j = g_j^{s_j}$  606

## Optional Randomization by Authorities

In sequence, each authority performs verifiable shuffle (of pairs) on  $K$  using  $(g, C = g^C)$  as the shuffle commitment, and returns the shuffled set,  $K'$ , along with the shuffle verification transcript,  $T(K, K', G, C)$

If the verification transcript is correct. Registration Server performs the substitution

$$K = K'$$

and stores the previous values, along with the shuffle verification transcript for audit purposes.

(This can be performed as part of initialization, and/or, at any intermediate stage of anonymous certificate distribution.)

## Anonymous Certificate Request and Generation Phase (each registrant in turn)

### Registrant

Generate Request

anonymous authentication request

### Registration Server

Retrieve H

$H$  (contains registrant's public key)

- 1) Select subset  $M \subset H$  of size  $k' \leq k$  and set  $M' = H - M$
- 2) Compute shuffle,  $H'$ , of  $M$  and verification transcript
- 3) Generate zero knowledge proof,  $P$  that registrant knows exponent  $s$  such that  $(g'_j)^s = h'_j \in H'$  for specified index,  $1 \leq j \leq k'$
- 4) Generate PKI Certificate Request with "random identifying information"
- 5) Safely store private key corresponding to this certificate request

$T(M, H', g, C), P$

$R$  = PKI Certificate Request

- 1) Check shuffle verification transcript
- 2) Check  $P$
- If both checks pass
- 3) Set  $K = J \cup M' \cup (H' - \{(g'_j, h'_j)\})$   
 $k = k - 1$
- 4) Store  $T(M, H', g, C)$  for audit purposes
- 5) Digitally sign  $R$  thereby creating PKI Certificate,  $\Omega(R)$
- Else, if any check fails

$\Omega(R)$

deny request

Loop to beginning of this phase (ready for next anonymous authentication request)

TOP SECRET